Child-Routing:-

app-routing.module.ts

```
import{NgModule, Component}from '@angular/core';
import{Routes, RouterModule}from '@angular/router';
import{student Component}from './student/student.component';
import { studentDetailsComponent}from './student-detail/student-details.compont';
import{ StudentRegistration Component}from './studentregistration/student
                                        registration.component';

const routes: Routes = [
{ path: 'student',
  children: [{ path: '', component: StudentComponent},
  { path: 'studentDetails', component: StudentDetailsComponent},
  {path: 'studentregis', component: StudentRegistration Component}]
 },
];
@NgModule({
  imports:[RouterModule.forRoot(routes)]
  exports:[RouterModule]
})
export class AppRoutingModule{ }
```

* In order to make use of nested component i.e., localhost: 4200/student/studregis

* student is the parent component sturegis is the child component.

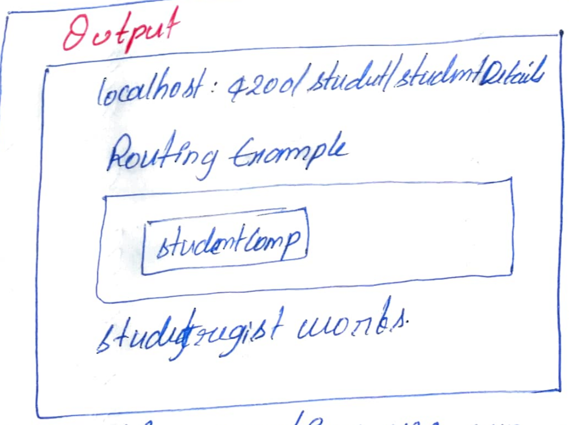
student.component.html

```
<p>Student works</p>
<body>
 <h1>Student Component</h1>
 <div>
 <a [routerLink]="['/studentDetails']">
          studentCompDetails</a>
<a [routerLink]="['/studentreg']">
          StudentRegistration</a>
 </div>
</body>
```

app.component.html

```
<body>
 <div> <h1>Routing Example</h1>
 <a [routerLink]="['/student']">
          studentcomp</a>
 </div>
</body>
<router_outlet></router-outlet>
```

Output



```
localhost: 4200/student/studentDetail
Routing Example
studentComp
studentregist works.
```

Note:- Using routing we can re-direct from one component to another component without refreshing. It is not compulsary to use children array

Service

- Syntax to create service

```
ng g s service
```

- Services are piece of code that are used to perform a specific task
- A service can contain a value or function or combinations of both
- Services are injected into application using dependency injection mechanism.
- Services prevents us from writing the same code at multiple sections of our application
- The best solution is to write store, and interact services are inject in application where we need it.
- Services provide, with data, and a communication channel b/w classes
- Service is a mechanism used to share the functionality b/w the components.

mynames.service.ts

```
import { Injectable } from '@angular/core';
@Injectable({
  providedIn: 'root'
})
export class MynameService
{
  constructor() { }
   display() {
   return "hello guys"
  }
}
```

app.component.ts.

```
import {Component} from '@angular/core';
import {MynameService} './myname.service';
@Component({
   selector:'app-root',
   templateUrl:'./app.component.html',
    styleUrls:['./app.component.css'],
     providers: [MynameService]
 })
 export class AppComponent{
   title='routing Example';
    text:string="";
  constructor(private mynameserve:MynameService){}
  ngOnInit():void{
        this.text=this.mynameserve.display()
   }
}
```